Efficient Ontology Meta-Matching Based on Metamodel-assisted Compact MOEA/D

Xingsi Xue

College of Information Science and Engineering
Fujian Provincial Key Laboratory of Big Data Mining and Applications
Fujian University of Technology
No.3 Xueyuan Road, University Town, Minhou, Fuzhou, Fujian, 350118, China
Corresponding author: jack8375@gmail.com

Pei-Wei Tsai

Fujian Provincial Key Laboratory of Big Data Mining and Applications
National Demostration Center for Experimental Electronic Information and Electronical Technology Education
Fujian University of Technology
No.3 Xueyuan Road, University Town, Minhou, Fuzhou, Fujian, 350118, China
peri.tsai@gmail.com

Guirong Feng

School of Electrical Engineering
Xidian University
No.2 South Taibai Road, Xian, Shaanxi, 710071, China
guirfeng@163.com

ABSTRACT. *In this paper we propose an improved Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D), i.e. metamodel-assisted compact MOEA/D, to efficiently solve the ontology meta-matching problem. Particularly, we dedicate to determine the optimal ontology alignment by tuning four different basic similarity measures, i.e. Syntactic Measure, Linguistic Measure, Taxonomy-based Measure and Instance-based Measure. The experimental results show that our proposal can significantly improve the efficiency of MOEA/D based approach without sacrificing the quality of the ontology alignment, and the results obtained by our approach are better than state-of-the-art ontology matching systems.*
Keywords: Ontology meta-matching, Metamodel, Compact MOEA/D

1. **Introduction.** Ontology is the main component of Semantic Web to solve the semantic heterogenous problem. However, due to the human's subjectivity, ontologies in the same domain could have various ways of defining domain entities, which rises semantic heterogeneous problem to a higher level. Ontology matching is a ground technology to solve this problem by determining the correspondences between heterogeneous entities. In recent years, although lots of ontology matching systems have been proposed to determine the entity correspondences between two ontologies, there're still many open problems remaining unsolved. One of them is how to efficiently tune the parameters, e.g. the weights and threshold, to aggregate different similarity measures' matching results and obtain an optimal ontology alignment, which is so called ontology meta-matching problem.

Since its beginning, Evolutionary Algorithms (EAs) are appearing as the most suitable methodology to address the ontology meta-matching problem [1]. However, most EA based meta-matching technologies use a single objective to evaluate the alignment quality during the generation process, even though a suitable computation of parameters could be better performed by evaluating the right compromise among different objectives involved in the meta-matching process. Multi-Objective Evolutionary Algorithms (MOEAs) are emerging as a state-of-the-art methodology to solve the ontology meta-matching problem. In addition, for dynamic applications, it is necessary to perform the combination of similarity measures and system self-tuning at run time. Therefore, in addition to quality of the aligning results, the efficiency of the matching process is of great importance especially. In this paper we propose an improved Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) [2], i.e. metamodel-assisted compact MOEA/D, to efficiently solve the ontology meta-matching problem. Particularly, we dedicate to determine the optimal ontology alignment by tuning four different basic similarity measures, i.e. Syntactic Measure, Linguistic Measure, Taxonomy-based Measure and Instance-based Measure.

The rest of the paper is organized as follows: section 2 presents the metamodel based on Gaussian Random Field Model; section 3 gives the details of metamodel-assisted compact MOEA/D; section 4 shows the experimental results and analysis; finally, section 5 draws the conclusions.

2. Metamodel based on Gaussian Random Field Model.

Metamodel is used to model a mathematical relationship, e.g. approximate multivariate function, between the points that have already been evaluated. More precisely, given a set of points $x^1, x^2, ..., x^n \in R^n$, and evaluations of the objective functions at these points $y^1 = f(x^1), ..., y^n = f(x^n)$, the metamodel can be used to compute an approximation $\vec{f}(x) \approx f(x)$ for any point $x \in R^n$ in a time that is considerably faster than the precise evaluations.

Gaussian Random Field Model (GRFM) is a particular type of interpolation model which can not only predict the objective function value but also provide a measure of confidence for its prediction. Actually, the output of GRFM includes both the mean value and standard deviation for an one-dimensional Gaussian distribution which represents the likelihood for different realizations of outcomes to represent a precise function evaluation. In this work, the evaluation function is thought to be the realization of a random process with a spatial index. The main assumption is that the random variables are correlated by a distance based correlation function, and we utilize the following Gaussian product kernel [3] as the correlation function:

$$c(\theta_1, \theta_2, ..., \theta_d) = \prod_{i=1}^{n} exp(-\theta_i \cdot |x_i - x_i'|^2) \tag{1}$$

where $\theta_i, i = 1, 2, ..., d$ denote correlation parameters. First, the free parameters, i.e. the θ values, are estimated by means of the maximum likelihood method. Then the conditional distribution of the random field for the new point x' is computed. Since the random field is Gaussian, this distribution is a one-dimension conditional Gaussian distribution $\mathcal{F}(x'|X, y)$, the mean value of which will serve as the predictor $\hat{y}$, whereas the standard deviation serves as the confidence measure $\hat{s}(x')$.

The quantity of the training points, i.e. the solutions need precise evaluation, is the main factor that determines the time consumption and especially for large training sets this is very time consuming. Therefore, it is recommended to use only the minimum necessary subset of the total number of training points available. A well performed heuristic

is to use a small number of training points which are closest, with regard to the Euclidian metric, to each new point x' and train a locally new valid metamodel. In our work, the value of the neighborhood size has been set to 3, and any further increase in 3 seems to slightly improve the results, but at the same time, increases the computation time significantly.

3. Metamodel-assisted Compact MOEA/D.

3.1. The Multi-Objective Optimal Model for Ontology Meta-matching Problem.
In this work, an ontology O is defined as $O = (C, P, I, \Lambda, \Gamma)$ [4], where C, P, I, Λ, Γ are respectively referred to the set of classes, properties, instances, axioms and annotations. In addition, an ontology alignment A between two ontologies is a correspondence set and each correspondence inside is a 4-tuples $(e, e', conf, rel)$, where e and e' are respectively the entities of two ontology, $conf \in [0, 1]$ is a confidence value for the correspondence between e and e' and rel is the relation of equivalence between e and e'.

Supposing the golden alignment is one to one, i.e. one entity in source ontology is matched with only one entity in target ontology and vice versa, based on the observations that the more correspondences found and the higher mean similarity values of the correspondences are, the better the alignment quality is [5], we propose the the following ontology alignment quality measures:

$$R(A) = \alpha \times MC(A) + (1 - \alpha) \times \frac{\sum_{i=1}^{|A|} \delta_i}{|A|} \tag{2}$$

$$P(A) = \alpha \times MR(A) + (1 - \alpha) \times \frac{\sum_{i=1}^{|A|} \delta_i}{|A|} \tag{3}$$

$$F(A) = \frac{2 \times R(A) \times P(A)}{R(A) + P(A)} \tag{4}$$

where $|A|$ is the number of correspondences in A, MC and MR are respectively the functions of calculating A's MatchCoverage and MatchRatio [6], δ_i is the similarity value of the ith correspondence in A, and α is a tuning parameter used to tradeoff the ontology alignments characterized by high precision (with the decreasing of α) or high recall (with the increase of α). In this work, α is set to 0.2 by referring to the work in [4]. On this basis, the optimal model of ontology matching problem is defined as follows:

$$\begin{cases} max & (R(X), P(X)) \\ s.t. & X = (x_1, x_2, ..., x_n, x_{n+1})^T \\ & \sum_{i=1}^{n} x_i = 1 \\ & x_i \in [0, 1], i = 1...n + 1 \end{cases} \tag{5}$$

where the decision variable X represents the parameter set, i.e. the weights for aggregating various similarity measures (x_i, $i = 1...n$), and a threshold (x_{n+1}) for filtering the aggregated alignment, used to obtain the final alignment.

3.2. Decomposition.
Since the two objectives in our work are maximizing $R()$ and $P()$, the ith decomposed single objective can be defined as follows:

$$max\left\{ \frac{R() \cdot P()}{\alpha_i \cdot R() + \beta_i \cdot P()} \right\} \tag{6}$$

where $\alpha_i + \beta_i = 1$, and assuming that N is the number of decomposed problems, $(\alpha_1, \beta_1) = (0, 1), (\alpha_i, \beta_i) = (\frac{i-1}{N-1}, \frac{N-i}{N-1}), i = 2, 3, ..., N - 1, (\alpha_N, \beta_N) = (1, 0)$. If the weight vectors $\lambda_i = (\alpha_i, \beta_i)$ and $\lambda_j = (\alpha_j, \beta_j)$ are close to each other, the ith single objective problem and

the jth one are neighbors, any information about the neighbors of the ith single objective problem should be helpful for optimizing the ith single objective problem.

3.3. **Elitism strategy.** For the single objective sub-problem that maximizes the $R()$ (or $P()$), the solution with higher $R()$ (or $P()$) is better, and if two solutions' $R()$ (or $P()$) is equal, the one with higher $P()$ (or $R()$) will outperform. With respect to the sub-problem that maximizes the $F()$, the solution with higher $F()$ is better, and if two solutions have the same $F()$, and we will adopt the max-min approach to determine a better solution, i.e., supposing solutions x_1 and x_2 have the highest $F()$, and their $R()$ and $P()$ values are denoted by $f_r(x_i)$ and $f_p(x_i)$, respectively, for $i = 1, 2$, and then we select the solution by the max-min approach as follows: $x_j = \arg_i \max\{\min\{f_r(x_i), f_p(x_i)\}\}$

3.4. **The detailed procedure of metamodel-assisted compact MOEA/D.** In this work, three representative single objective sub-problems with the weight vector $(1,0)$, $(0.5,0.5)$ and $(0,1)$ are selected, which represent the single objective optimization sub-problems of maximizing $R()$, $P()$ and $F()$, respectively. After that, we use three PVs to respectively represent the populations for solving these representative sub-problems. In particular, the PV here represents the population that consists of the solutions of its corresponding representative sub-problem and this problem's neighbor sub-problems. Finally, these PVs help each other in the process of determining three representative solutions, which is shown as following. Here, we mark three representative sub-problems of maximizing $R()$, $P()$ and $F()$ with the symbols P_r, P_p and P_f, respectively, and three PVs for P_r and P_p and P_f with the symbols PV_r, PV_p and PV_f, respectively.

   Input:
- O_1 and O_2: two ontologies to align;
- num: the length of PV;
- $max\_generation$: maximum number of generations;
- p_c: the uniform crossover probability.

Output: three solutions with best $R()$, $p()$ and $F()$, respectively. In the absence of explicit decision makers preferences, the knee regions could represent the decision makers preferences themselves [7]. Thus, in this work, three knee solutions, i.e. those with best $R()$, $p()$ and $F()$ respectively, are selected as the represented solutions from pareto front.

   Step 1) Initialization:

      Step 1.1) Set the neighbor sub-problem of P_r and P_p as P_f, and the neighbor sub-problems of P_f as P_r and P_p;

      Step 1.2) Initialize PV_r, PV_p and PV_f by setting all the probabilities inside as 0.5;

      Step 1.3) Using PV_r and PV_p and PV_f to generate the elites, which are marked with symbols $elite_r$, $elite_p$ and $elite_f$, for P_r, P_p and P_f, respectively;

      Step 1.4) Evaluate $elite_r$, $elite_p$ and $elite_f$ and insert results into Database D;

   Step 2) Update:

      Step 2.1) Update PV_r, PV_p and PV_f respectively:

      For example, when updating PV_r (the procedures of updating P_p and P_f is similar to it):

      1. generate one individual a through PV_r;

      2. Evaluate a with metamodel derived from D and insert the result into D;

      3. $[winner, loser] = compete(elite_r, a)$;

      4. if a==winner then $elite_r = a$;

      5. for $i = 1 : n$ do

      6. if $winner[i] != loser[i]$ then

TABLE 1. A brief description of the benchmark in OAEI 2016.

Case ID	Brief Description
1XX	Two identical ontologies with different constraints
2XX	Two Ontologies with different lexical, linguistic or structure features.
3XX	Two real world's ontologies

7. if $winner[i] == 1$ then
8. $PV_r[i] = PV_{mc}[i] + 1/num$;
9. else
10. $PV_r[i] = PV_{mc}[i] - 1/num$;

Step 2.2) Update PV_r, PV_p and PV_f mutually:
For P_r (or P_p), PV_{temp} is generated by applying the p_c based uniform crossover operator on the PV_r (or P_p) and its neighbor sub-problem's probability vector PV_f. Then generate an individual a through PV_{temp}, and try to update the P_r and PV_f according to the steps presented in Step 2.1;

For P_f, PV_{temp} is generated through applying the uniform crossover operator between PV_r and PV_p, which are its neighbor sub-problems' PVs. Then generate an individual a through PV_{temp}, and try to update the P_f according to the steps given in Step 2.1;

Step 3) Stopping Criteria:
If the stopping criteria is satisfied, then stop and output three solutions with best $R()$, $P()$ and $F()$, respectively; otherwise, go to Step 2).

In the evolutionary process (step 2), we first update PV_r, PV_p and PV_f respectively (step 2.1), which is equivalent to the process of updating the solutions of P_r, P_p and P_f through their respective neighbor sub-problems' solutions. Then, in the step 2.2, we update PV_f, PV_p and PV_f mutually, which is equal to update the solutions of P_f, P_p and P_f through their shared neighbor sub-problems' solutions, i.e. using the information of a PV to help its neighbor PVs.

4. **Experimental Results and Analysis.** In order to study the effectiveness of our proposal, we have exploited the benchmark provided by the Ontology Alignment Evaluation Initiative (OAEI) 2016 [8], which is commonly used for experimentation about ontology matching problem and the brief description of it is given in Table 1, where 1XX, 2XX and 3XX represent the test case ID begins with 1, 2 and 3.

4.1. **Experiment Configuration.** In this work, the similarity measures used are as follows:
- SMOA distance [9] (Syntactic Measure),
- Linguistic distance [10] (Linguistic Measure),
- Taxonomy distance [11] (Taxonomy-based Measure),
- N-Gram distance [12] and Jaro Winkler distance [13] with similarity upPropagation algorithm [14] (Instance-based Measure).

The parameters used by compact MOEA/D are listed as follows:
- Three representative decomposing weight vectors: (1,0), (0.5,0.5) and (0,1),
- Crossover probability for PVs: 0.6,

TABLE 2. Comparison of the alignments of best f-measure obtained by OAEI 2016 participants, MOEA/D based approach and our approach.

Matching System	recall	precision	f − measure
AML	0.24	1.00	0.38
CroMatcher	0.83	0.96	0.89
Lily	0.83	0.97	0.89
LogMap	0.39	0.93	0.55
LogMapLt	0.50	0.43	0.46
PhenoMF	0.01	0.03	0.01
PhenoMF	0.01	0.03	0.01
PhenoMP	0.01	0.02	0.01
XMap	0.40	0.95	0.56
LgoMapBio	0.24	0.48	0.32
MOEA/D based approach	0.91	0.95	0.93
our approach	0.89	0.94	0.91

- maximum number of generations: 300.

The parameters used by MOEA/D are listed as follows [15]:

- Decomposed sub-problems' number: 21,
- Population size: 201,
- Selection probability: 0.8,
- Crossover probability: 0.98,
- Mutation probability: 0.05,
- maximum number of generations: 300.

We set the number of neighbors of each weight vector in MOEA/D as 5. With regard to the population size, since we want the weight vector (0.5,0.5) in MOEA/D to be included by $\{(0, \frac{N-1}{N-1}), (\frac{1}{N-1}, \frac{N-2}{N-1}), ..., (\frac{N-2}{N-1}, \frac{1}{N-1}), (\frac{N-1}{N-1}, 0)\}$, which makes sure the decomposed single objective problems contain the value of f-measure as one of its members, we need to set N as an odd number.

4.2. **Results and analysis.** In order to compare the quality of our proposal with other approaches, we evaluate the obtained alignments with traditional recall, precision and f-measure, and the results in Tables are the mean values in thirty time independent executions. In particular, Table 2, Table 3 respectively show the quality of alignment and runtime by OAEI 2016 participants [16], MOEA/D based approach and our approach.

It can be seen from Table 2 that our approach can obtain much better ontology alignment than OAEI 2016 participants in terms of f-measure, and the quality of our approach is very close to that of MOEA/D based approach. In Table 3, our approach's runtime is much less than all the other ontology matching systems, which is 32% of that by MOEA/D based approach. In addition, the total memory consumption of our approach is approximately 42 gigabytes, which is much less than MOEA/D based approach's 164 gigabytes. Thus, we can draw the conclusion that metamodel-assisted compact MOEA/D is able to efficiently solve the ontology meta-matching problem without sacrificing the alignment's quality, and the results obtained by our approach are better than the state-of-the-art ontology matching systems.

5. **Conclusion.** One of the challenges in ontology matching domain is how to select weights and thresholds in ontology matching process in order to aggregate the various

TABLE 3. Comparison of MOEA/D based approach and our approach in terms of total runtime (second)

Ontology Matching System	$runtime(second)$
AML	120
CroMatcher	1100
Lily	2211
LogMap	194
LogMapLt	96
PhenoMF	1632
PhenoMF	1743
PhenoMP	1833
XMap	123
LgoMapBio	54439
MOEA/D based approach	253
our approach	82

similarity measures to obtain a satisfactory alignment, which is so called ontology meta-matching problem. In this paper, we present a metamodel-assisted compact MOEA/D, in the whole similarity aggregation step of meta-matching process, to provide the diverse global non-dominated solutions of weights and thresholds for meta-matching system to meet the diverse requirements of alignment. The experimental results show that our proposal is able to highly improve the efficiency of determining the optimal alignments through MOEA/D based approach, and the quality of the alignments obtained is better than the state-of-the-art ontology matching systems.

REFERENCES

[1] J. Martinez-Gil, E. Alba and J. F. Aldana-Montes , Optimizing ontology alignments by using genetic algorithms, *Nature Inspired Reasoning for the Semantic Web (NatuReS2008)*, vol.419, pp.31-45, 2008.

[2] Q. Zhang and H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, vol.11, no.6, pp.712-731, 2007.

[3] J. Sacks, W. J. Welch , T. J. Mitchell and H. P. Wynn , Design and analysis of computer experiments, *Statistical science*, pp.409–423, 1989.

[4] G. Acampora, V. Loia and A. Vitiello , Enhancing ontology alignment through a memetic aggregation of similarity measures, *Information Sciences*, vol.250, pp.1-20, 2013.

[5] J. Bock and J. Hettenhausen , Discrete particle swarm optimisation for ontology alignment, *Information Sciences*, vol.192, pp.152-173, 2012.

[6] X. Xue and Y. Wang , Optimizing ontology alignments through a Memetic Algorithm using both MatchFmeasure and Unanimous Improvement Ratio, *Artificial Intelligence*, vol.223, pp.65-81, 2015.

[7] S. Bechikh , L. B. Said and K. Ghedira , Searching for knee regions of the Pareto front using mobile reference points, Soft Computing, vol.15, no.9, pp.1807-1823, 2011.

[8] Ontology Alignment Evaluation Initiative (OAEI) 2016, Available at http://oaei.ontologymatching.org/2016/, Accessed on 15 January, 2017.

  [9] G. Stoilos , G. Stamou ,S. and Kollias , A string metric for ontology alignment, *Proceedings of 4th International Semantic Web Conference*, pp.623-637, 2005.

[10] G. A. Miller , Wordnet: A lexical database for english, *Communications of the ACM*, vol.38, no.11, pp.39C41, 1995.

[11] S. Melnik , H. Garcia-Molina and E. Rahm , Similarity flooding: A versatile graph matching algorithm and its application to schema matching, *18th International Conference on Data Engineering*, pp.117-128, 2002.

[12] Kondrak G., N-gram similarity and distance, *International Symposium on String Processing and Information Retrieval*, pp.115-126, 2005.

[13] M. Rajabzadeh, S. Tabibian , A. Akbari, and Nasersharif B., Improved dynamic match phone lattice search using viterbi scores and Jaro Winkler distance for keyword spotting system, *16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP)*, pp.423-427, 2012.

[14] X. Xue, Y. Wang and J. Hou , Ontology Alignment based on Instances using NSGA-II, *Journal of Information Science*, vol.41, no.1, pp.58-70, 2015.

[15] X. Xue, Y. Wang and W. Hao , Using MOEA/D for Optimizing Ontology Alignments, *Soft Computing*, vol.8, no.18, pp.1589-1601, 2014.

[16] M. Achichi, M. Cheatham, Z. Dragisic, et al., Results of the Ontology Alignment Evaluation Initiative 2016, *11th ISWC workshop on ontology matching*, pp.73-129, 2016.